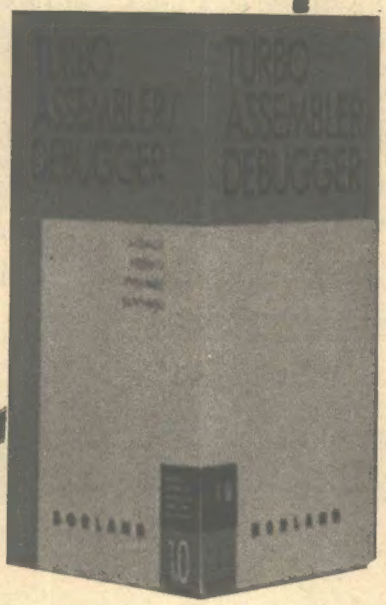
เทอร์โบ ดีบักเกอร์ 1.0



TWT

เมื่อประมาณเดือนสิงหาคมปีที่แล้ว บอร์แลนด์ได้ออกผลิตภัณฑ์ใหม่ตระกูลเทอร์โบมาถึง 4 ตัวด้วยกัน สำหรับหลาย ๆ ท่านที่เป็นแฟนของบอร์แลนด์คงสนุกกับเทอร์โบปาสคาลและเทอร์โบซีเวอร์ชันใหม่ที่มีความสามารถเพิ่มขึ้นจากเดิมอีกมากทีเดียว แต่ที่โดดเด่นเห็นจะหนีไม่พ้นส่วนของดีบักเกอร์ ซึ่งสามารถดีบักโปรแกรมได้ในระดับซอร์สโค้ด อย่างไรก็ตามถ้าโปรแกรมมีความยุ่งยากมากเช่น มีโครงสร้างของข้อมูลที่ซับซ้อนหรือโปรแกรมที่มีขนาดใหญ่ ดีบักเกอร์ที่อยู่ในเทอร์โบปาสคาลหรือเทอร์โบซีเองก็ไม่ได้ช่วยอะไรเรามากนัก วันนี้เราจะมารู้จักกับดีบักเกอร์อีกตัวหนึ่งซึ่งเป็นหนึ่งในสี่ที่ออกใหม่ของบอร์แลนด์ นั่นคือ เทอร์โบดีบักเกอร์ 1.0 ซึ่งบอร์แลนตั้งใจทำให้สะดวกแก่ผู้ใช้มากที่สุด เทอร์โบดีบักเกอร์จึงใช้เป็นเมนูทั้งหมด เปรียบเทียบกับโค้ดวิวของไมโครซอฟต์ที่มีความยุ่งยากมากกว่า เพราะใช้ทั้งคำสั่งและเมนู เทอร์โบดีบักเกอร์มี hot keys มากกว่า 200 keys รวมทั้งคำสั่งจากเมนูหลักและเมนูภายใน (local menu) ในส่วนของหน้าจอมีแถบเมนูอยู่บนสุด ซึ่งแสดงฟังก์ชันพื้นฐาน เช่น การจัดการกับไฟล์ การตั้งเบรคพอยต์ และการรันโปรแกรมเป็นต้น ส่วนบรรทัดล่างสุดแสดงฟังก์ชันคีย์ซึ่งเป็น context-sensitive (หมายถึงฟังก์ชันคีย์ซึ่งเปลี่ยนไปตามหน้าต่างในขณะนั้น) เมื่อกดปุ่ม Alt บรรทัดล่างจะแสดงปุ่มที่ใช้ร่วมกับปุ่ม Alt ถ้ากดปุ่ม Ctrl จะแสดงคำสั่งภายใน (local command) ซึ่งเป็นปุ่มตัวอักษรในคำสั่งที่ใช้ร่วมกับปุ่ม Ctrl บอร์แลนด์ยังแบ่งหน้าต่างออกเป็นเพน (pane) ซึ่งสามารถย้ายจากเพนหนึ่งไปยังอีกเพนหนึ่งด้วยปุ่ม tab ทำให้ดูรายละเอียดได้มากขึ้นเช่นหน้าต่างของการดูซีพียู ซึ่งแบ่งออกเป็น 5 เพน คือ เพนของรหัส เพนของรีจิสเตอร์ เพนของแฟล็ก เพนของสแต็ก และเพนของข้อมูล อีกทั้งการแอคตีฟหน้าต่างที่เปิดอยู่แล้วด้วยการกดปุ่ม Alt พร้อมกับหมายเลขหน้าต่างก็สะดวกกว่าการเลือกจากรายการที่มีอยู่เดิม (history lists)

สภาวะแวดล้อมของการดีบัก

เทอร์โบดีบักเกอร์มีสิ่งสนับสนุนที่สร้างขึ้นสำหรับฮาร์ดแวร์ภายใต้สภาวะแวดล้อมที่แตกต่างกัน 4 อย่างคือ โหมดของ 8086 โหมด virtual ของ 80386 การดีบักระยะไกล (remote debugging) และการดีบักโดยมีฮาร์ดแวร์เป็นตัวช่วย เราสามารถให้เทอร์โบดีบักเกอร์เอ็กซีคิวต์โปรแกรมทีละบรรทัดผ่านเข้าไปในฟังก์ชันหรือผ่านไปโดยเอ็กซีคิวต์ทีเดียวตลอดฟังก์ชันเลยก็ได้ ซึ่งสามารถทำได้ในระดับซอร์สโค้ดและระดับแอสเซมบลีสำหรับทุกโหมด

สำหรับการดีบักบนเครื่อง 80386 โดยใช้ลักษณะของหน่วยความจำแบบ virtual จะต้องติดตั้งไดรเวอร์ TDH 386.SYS ในไฟล์ CONFIG.SYS เสียก่อนแล้วจึงรัน TD386 TD386 จะใช้หน่วยความจำที่ขยายออกไป (extended memory) นี้ และให้โปรแกรมที่จะถูกดีบักโหลดและรันในหน่วยความจำจริง (virtual memory) ซึ่งทำให้โปรแกรมรันอยู่ในสภาวะแวดล้อมและเงื่อนไขที่เป็นจริง ดีไวซ์ไดรเวอร์ของ 386 ยังยอมให้เราตั้งเบรคพอยต์เป็นแบบฮาร์ดแวร์ได้ด้วยสำหรับใช้กับการเฟรชคำสั่ง การอ่านหน่วยความจำ และการเขียน/อ่านซึ่งกระทำกับหน่วยความจำที่ใดที่หนึ่งโดยเฉพาะ

ในกรณีที่ไม่มีพีซีที่เป็น 80386 หรือหน่วยความจำมีจำกัด เรามีทางเลือกที่ดีกว่าคือ การใช้ TDREMOTE ซึ่งเป็นโปรแกรมอินเตอร์เฟสใช้สำหรับ

การดีบักระยะไกล โดยรันเทอร์โบดีบักเกอร์บนเครื่องหนึ่ง และรัน TDREMOTE กับโปรแกรมที่ถูกดีบักบนอีกเครื่องหนึ่ง ซึ่งการแยกจากกันนี้สามารถป้องกันความเสียหายที่จะเกิดแก่ดีบักเกอร์ได้ TDREMOTE ต้องการหน่วยความจำเพียง 15 กิโลไบต์เท่านั้น เพื่อให้ผู้ใช้สามารถดีบักโปรแกรมขนาดใหญ่ได้ (ดูขนาดของโปรแกรมที่ใหญ่ที่สุดที่ถูกดีบักได้ เปรียบเทียบระหว่างเทอร์โบดีบักเกอร์และโค้ดวิวจากตารางที่ 1) TDREMOTE และเทอร์โบดีบักเกอร์ติดต่อกันโดยผ่านพอร์ตอนุกรมซึ่งไม่ต้องใช้โมเดมเลย

TDRF เป็นอีกโปรแกรมหนึ่งซึ่งถูกใช้สำหรับดีบักเกอร์เพื่อช่วยในการถ่าย (transfer) ไฟล์ระหว่างเครื่องที่รันเทอร์โบดีบักเกอร์และเครื่องที่รันโปรแกรมที่ถูกดีบัก TDRF สามารถถ่ายไฟล์ด้วยอัตรา 9600, 40,000 หรือ 115,000 บิตต่อวินาที(bps) และยังสามารถใช้ฟังก์ชันปกติของดอสได้ เช่น การลบไฟล์ การเปลี่ยนชื่อไฟล์ การสร้างหรือการลบไดเรคทอรี

เทอร์โบดีบักเกอร์สามารถใช้ EMS 3.2 หรือ 4.0 เพื่อเก็บตารางของสัญลักษณ์ (symbols table) และคอยติดตามจัดการกับหน่วยความจำที่เพิ่มขึ้นมา (expanded memory) ซึ่งถูกใช้โดยโปรแกรมที่ถูกดีบัก แต่จะถูกใช้สำหรับตารางของสัญลักษณ์เท่านั้น จะไม่มีส่วนที่เอ็กซีคิวต์ได้ถูกโหลดลงใน EMS

ในกรณีที่เป็นเครื่องพีซี 80386 เทอร์โบดีบักเกอร์จะรันโปรแกรมที่ถูกดีบักในที่ว่าง 1 เมกะไบต์ (ที่ว่างสำหรับโปรแกรม 640 กิโลไบต์) และใช้ฮาร์ด-

บริษัทผู้ผลิต	Borland	Microsoft
ชื่อโปรแกรม	Turbo Debugger	Code View
เวอร์ชัน	1.0	2.2
การดีบักโปรแกรม		
คอมพายล์ด้วย/Zi (TDCONVRT)	มี	มี
โดยใช้ MAP files (TDMAP)	มี	–
สนับสนุนภาษาอื่นด้วย .MAP files	มี	–
OS/2	–	มี
Microsoft Windows	–	มี
Turbo C 2.0 and Turbo Pascal 5.0	มี	–
ความสามารถอื่น ๆ ในการดีบัก		
ดู รีจิสเตอร์ 32 บิต	มี	มี
ใช้ความสามารถในการดีบักของฮาร์ดแวร์ของ 80386	มี	–
มีข้อมูลของการสนับสนุนการดีบักด้วยฮาร์ดแวร์ของ 80386	มี	–
มี remote debugging	มี	–
ใช้คำสั่งที่เป็น macro หรือ file	Macro[1]	file
มีความสามารถในการบันทึก output (log output)	มี	มี[2]
ดูโครงสร้างข้อมูลในรูปแบบของโปรแกรมเดิม (source program)	มี	–
สามารถใช้คำสั่งโดยใช้ mouse ช่วยได้	–	มี
ใช้ expand memory	มี	มี
รายงานการใช้ EMS ของตัว Turbo Debugger เองและโปรแกรมที่ถูกดีบักด้วย.	มี	–
สามารถค้นหาคำสั่งในโปรแกรมแอสเซมบลีได้	มี	–
ขนาดสูงสุดของโปรแกรมและ DOS (ในหน่วย กิโลไบต์)		
8088, 8086, หรือ 80286	411	410
80386	680	410
Remote PC	625	–

1. คำสั่ง macro ไม่สามารถ save ในระหว่างทำการดีบักโปรแกรม
2. ผลลัพธ์ที่ออกทางหน้าจอจะถูกส่งไปเข้าไฟล์โดยไม่ปรากฏให้ผู้ใช้เห็น

ตารางที่ 1 เปรียบเทียบขีดความสามารถของเทอร์โบดีบักเกอร์และโค้ดวิว

แวร์ของ 80386 ในการกันหน่วยความจำป้องกันโปรแกรมที่ถูกดีบักไม่ให้ถูกรบกวนจากสภาวะแวดล้อม

ขณะที่เทอร์โบดีบักเกอร์ใช้สำหรับคอมพายล์และแอสเซมเบลอร์โปรแกรมของบอร์แลนด์นั้น เรายังสมารถดีบักโปรแกรมที่เขียนขึ้นโดยใช้ไมโครซอฟต์-ในระดับซอร์สโค้ดได้อีกด้วย โดยใช้โปรแกรม TDCONVRT ซึ่งมีมากับเทอร์โบดีบักเกอร์ TDCONVRT จะทำการแปลงไฟล์ .EXE ซึ่งถูกคอมพายล์สำหรับโค้ดวิวเป็นไฟล์ .EXE ซึ่งอยู่ในรูปแบบของบอร์แลนด์

เทอร์โบดีบักเกอร์สามารถใช้ได้กับคอมพายเลอร์ และลิงค์เกอร์เกือบทุกตัวที่ให้รายละเอียดเป็นไฟล์ .MAP โดยใช้ยูทิลิตี้ TDMAP เพื่อเพิ่มข้อมูลให้กับเทอร์โบดีบักเกอร์ โดยเพิ่มลงไปในไฟล์ .EXE เทอร์โบดีบักเกอร์สามารถใช้ภายใต้ดอสเท่านั้น ไม่สามารถใช้ภายใต้โอเอสทู (OS/2) และโปรแกรมซึ่งคอมพายล์สำหรับไมโครซอฟต์วินโดว์ซึ่งตรงข้ามกับของไมโครซอฟต์ที่เพิ่มคุณลักษณะพิเศษให้กับโค้ดวิวเพื่อใช้กับโอเอสทูได้

ก่อนเริ่มต้นการดีบัก

เราควรติดตั้ง (INSTALL) พารามิเตอร์ต่าง ๆ เสียก่อน โดยใช้โปรแกรม TDINST ซึ่งเป็นโปรแกรมควบคุมพารามิเตอร์ต่าง ๆ ของเทอร์โบดีบักเกอร์ เราสามารถเปลี่ยนค่าเหล่านี้ได้ เช่น การเลือกสีของหน้าต่าง เลือกเอดิเตอร์ตัวที่เราชอบ เลือกไดเรคเทอรีของดีบักเกอร์ ยอมให้ใช้การดีบักแบบระยะไกลหรือการใช้ EMS หรือไม่ การเลือกใช้ภาษาเพื่อให้ถูกซินแทกซ์ (syntax) เพื่อใช้สำหรับการหาค่าของนิพจน์ และการเลือกการแสดงผลซึ่งถูกจัดการระหว่างหน้าจอของดีบักเกอร์และหน้าจอของผู้ใช้ (หน้าจอที่ใช้แสดงเอาต์พุตของโปรแกรม) TDINST มีทางเลือกให้หลายทางสำหรับการจัดการกับหน้าจอซึ่งสลับไปมาระหว่างของดีบักเกอร์และผู้ใช้ ถ้าเครื่องของเราเป็นจอซึ่งมีหลายเพจ (page) เช่นจอแบบ CGA EGA และ VGA เทอร์โบดีบักเกอร์จะใช้การแสดงผลแบบแยกเพจ ทำให้เราสามารถสลับเพจได้ด้วยซอฟต์แวร์ซึ่งทำงานช้า แต่เป็นวิธีที่ดีที่สุด อีกวิธีหนึ่งที่ดีในการแสดงผลคือการใช้มอนิเตอร์ 2 ตัว ตัวหนึ่งใช้สำหรับดีบักเกอร์ อีกตัวหนึ่งใช้สำหรับแสดงผลของโปรแกรมซึ่งจะต้องใช้มอนิเตอร์คนละชนิดกัน ถ้าตัวหนึ่งเป็นมอนิเตอร์สีอีกตัวหนึ่งจะต้องใช้โมโนโครม

การดีบักระดับซอร์สโค้ดจะต้องใช้โปรแกรมที่คอมพายล์หรือแอสเซมเบลอร์แล้วตามด้วย option สำหรับของบอร์แลนด์จะต้องใช้ command line option /V หรือถ้าใช้จากเมนูจะต้องคอมพายล์โดยรวมข้อมูลของการดีบักเข้าไปในไฟล์ .OBJ ด้วย ส่วนของไมโครซอฟต์ที่ใช้โค้ดวิวเป็นดีบักเกอร์ต้องใช้ option /Zi แล้วใช้ TDCONVRT แปลงไฟล์ .EXE อีกทีหนึ่ง สำหรับโปรแกรมที่ถูกคอมพายล์และลิงค์อย่างเหมาะสมแล้ว เทอร์โบดีบักเกอร์จะแสดงโปรแกรมระดับซอร์สโค้ดโดยเริ่มต้นที่บรรทัดแรกที่สามารถเอ็กซีคิวต์ได้ ถ้าไม่เช่นนั้นแล้วมันจะแสดงโปรแกรมในระดับแอสเซมบลี หลังจากที่เรารู้จักเทอร์โบดีบักเกอร์พอสมควรแล้ว คราวนี้เรามาดูกันว่าเทอร์โบดีบักเกอร์ทำอะไรให้เราได้บ้างในการดีบักโปรแกรม

ความสามารถโดยสังเขป

เทอร์โบดีบักเกอร์เพียงตัวเดียวสามารถทำงานด้านดีบักโปรแกรมได้มีประสิทธิภาพมากกว่าดีบักเกอร์ที่มีอยู่ในตัวคอมพายเลอร์เองมาก เราสามารถใช้เทอร์โบดีบักเกอร์กับโปรแกรมที่เขียนด้วยภาษาซี ภาษาปาสคาล และภาษาแอสเซมบลี ไม่ว่าจะใช้คอมพายเลอร์ของบอร์แลนด์เองหรือของไมโครซอฟต์ก็ตาม เทอร์โบดีบักเกอร์สามารถช่วยให้เราหาที่ผิดและสาเหตุของความผิดพลาดนั้นได้เร็วขึ้น โดยการลดความเร็วในการเอ็กซีคิวต์โปรแกรม ทำให้เราสามารถตรวจสอบขั้นตอนของโปรแกรมทีละขั้น ตรวจดูค่าของตัวแปรและยอมให้เราเปลี่ยนค่าของตัวแปรเพื่อดูผลกระทบที่เกิดขึ้น ความสามารถต่าง ๆ รวมอยู่ในคำสั่งต่อไปนี้

TRACING ทำการเอ็กซีคิวต์โปรแกรมทีละบรรทัด

STEPPING ทำการเอ็กซีคิวต์โปรแกรมทีละบรรทัด และสามารถกระโดดข้ามฟังก์ชันหรือโพรซีเดอร์ที่มั่นใจแล้วว่าไม่มีที่ผิด เพื่อประหยัดเวลาในการดีบัก

VIEWING ทำการเปิดหน้าต่าง (WINDOW) เพื่อแสดงค่าต่าง ๆ เช่น ตัวแปร ค่าของตัวแปร เบรคพอยต์ ค่าในสแต็ก ตารางบันทึก ไฟล์ข้อมูล ซอร์สไฟล์ รหัสของซีพียู หน่วยความจำ รีจิสเตอร์ ข้อมูลของตัวประมวลผลทางคณิตศาสตร์ หรือเอาต์พุตของโปรแกรม

INSPECTING ตรวจดูอย่างละเอียดในกรณีที่เป็นตัวแปรที่มีโครงสร้างข้อมูลซับซ้อน เช่น อะเรย์ ยูเนียน เรคอร์ด หรือโครงสร้าง

CHANGING เปลี่ยนค่าของตัวแปรไม่ว่าแบบโกลบอลหรือโลคอล

WATCHING ดูการเปลี่ยนค่าของตัวแปรขณะที่โปรแกรมกำลังดำเนินไป

จากความสามารถดังกล่าวจะเห็นว่าเทอร์โบดีบักเกอร์ทำงานได้อย่างมากมาย แต่ก็มีบางสิ่งบางอย่างที่ไม่สามารถทำได้ เช่น เทอร์โบดีบักเกอร์ไม่มีเอดิเตอร์ในตัวมันเอง ที่จะแก้ซอร์สโปร-

แกรมให้เราได้ เราจะต้องใช้ local menu command เรียกเอดิเตอร์ที่เราถนัดมาใช้ ซึ่งเราได้กำหนดไว้ก่อนแล้วโดยการใช้โปรแกรม TDINST เทอร์โบดีบักเกอร์ไม่สามารถคอมพายล์โปรแกรมให้เราได้ เราต้องใช้คอมพายเลอร์ตัวเดิม เช่น เทอร์โบปาสคาล เทอร์โบซี

จุดเด่นของเทอร์โบดีบักเกอร์

อย่างที่ทราบแล้วว่าบอร์แลนด์ตั้งใจทำเทอร์โบดีบักเกอร์ให้สะดวกสบายแก่ผู้ใช้มากที่สุด นอกจากการใช้เป็นเมนูทั้งหมดแล้ว ยังมีสิ่งอำนวยความสะดวกอื่น ๆ อีกเช่น

- context-sensitive pop-up menus เช่น เราต้องการดูค่าตัวแปรตัวใดตัวหนึ่ง เราสามารถจะเลื่อนเคอร์เซอร์ไปที่ตัวแปรนั้น แล้วใช้ hot key เช่น Ctrl-W เทอร์โบดีบักเกอร์จะรู้ทันทีว่าเราต้องการดูตัวแปรตัวนั้น โดยเราไม่ต้องพิมพ์ชื่อตัวแปรลงไป

- สามารถเก็บรายชื่อที่พิมพ์เข้าไปแล้ว เช่นชื่อตัวแปรที่ต้องการค้นหา (search) โดยสามารถนำรายชื่อเดิมมาใช้ แก้ไขหรือพิมพ์รายชื่อเข้าไปใหม่

- สามารถใช้แมคโคร (MACRO) เพื่อเพิ่มความเร็วของชุดคำสั่ง หรือชุดของการใช้ปุ่มคีย์บอร์ด

- สะดวกสบายกับการใช้หน้าต่างที่สมบูรณ์แบบ

- สามารถใช้ online help ได้ทุกขณะ

ฮาร์ดแวร์และซอฟต์แวร์

เทอร์โบดีบักเกอร์สามารถรันบนเครื่อง IBM PC, XT, AT และ PS/2 และเครื่องที่คอมแพตทิเบิ้ลกับไอบีเอ็ม

รูปที่ 1 ไฟล์ต่าง ๆ ในเทอร์โบดีบักเกอร์

```
TURBO DEBUGGER/INSTALLATION
---------------------------
INSTALL  EXE  - Installs Turbo Debugger on your system
README   COM  - Program to display README file
TD       EXE  - Turbo Debugger Program
TD       OVL  - Turbo Debugger Program Overlay
HELPME!  DOC  - Text file with the answers to many common
                questions. Please read HELPME!.DOC before
                contacting Technical Support
README        - Last-minute information

HELP/UTILITIES
--------------
TDHELP   TDH  - Turbo Debugger Help File
TDINST   EXE  - Configuration Customization program
TDMAP    EXE  - Utility to extract debug information from .MAP
                files
TDUMP    EXE  - Utility to dump executable and object files
TDRF     EXE  - Remote file transfer utility
TDREMOTE EXE  - Remote debugging kernel
TD386    EXE  - Utility for 80386 virtual debugging
TDH386   SYS  - Device driver for TD386.EXE and the 80386 debug
                registers
TDSTRIP  EXE  - Utility to strip debug information off
                executable files
MANUAL   DOC  - Text file listing additions and corrections
                to the User's Guide

UTILITIES/EXAMPLES
------------------
TDCONVRT EXE  - Utility to convert CodeView information to Turbo
                Debugger information
TDPACK   EXE  - Utility to compress debug information
TDNMI    COM  - TSR to clear NMI every half second
UNPACK   COM  - Utility to unpack .ARC files
TDEXMPL  ARC  - Packed file that contains Turbo Debugger examples
TAEXMPL1 ARC  - Packed file that contains Turbo Assembler examples
TAEXMPL2 ARC  - Packed file that contains Turbo Assembler examples
CHAPXMPL ARC  - Packed file that contains Turbo Assembler examples
```

ที่ใช้ดอสเวอร์ชัน 2.0 ขึ้นไป โดยมีหน่วยความจำอย่างน้อย 384 กิโลไบต์ และควรมีฮาร์ดดิสก์ อย่างไรก็ตามถ้ามีเพียงดิสก์ไดรฟ์ 2 ตัวก็สามารถใช้งานได้ ไฟล์ต่าง ๆ ในเทอร์โบดีบักเกอร์มีดังรูปที่ 1

สำหรับผู้ที่ใช้ฮาร์ดดิสก์ให้เริ่มต้นที่แผ่น TURBO DEBUGGER/INSTALLATION โดยรันโปรแกรม INSTALL ซึ่งจะบอกรายละเอียดต่าง ๆ ในการติดตั้ง ส่วนผู้ที่ใช้ฟลอปปี้ดิสก์ให้เตรียมแผ่นเปล่าอีก 4 แผ่น เพื่อติดตั้งเทอร์โบดีบักเกอร์ แล้วรันโปรแกรม INSTALL เช่นกัน

เริ่มต้นการดีบัก

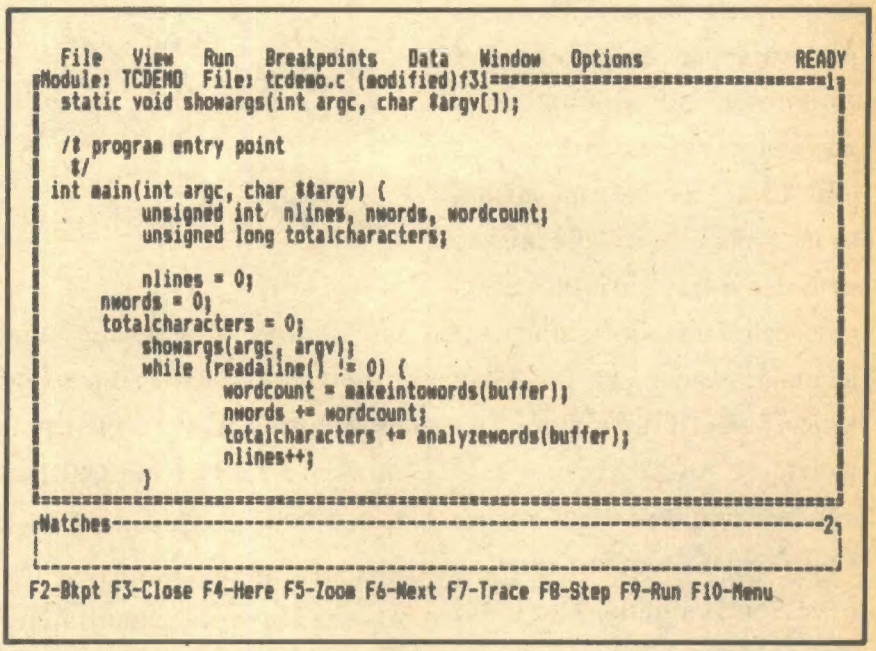
ตอนนี้เรามาลองใช้เทอร์โบดีบักเกอร์เพื่อดีบักโปรแกรมง่าย ๆ โปรแกรมหนึ่งซึ่งเขียนด้วยภาษาซี ชื่อ TCDEMO.C และ TPDEMO.PAS (สำหรับผู้ใช้ภาษาปาสคาล) ซึ่งมีอยู่ในแผ่น UTILITIES/EXAMPLES ตัวอย่างต่อไปนี้จะแสดงความสามารถเบื้องต้นของเทอร์โบดีบักเกอร์ดังต่อไปนี้

- แสดงการรันและหยุดโปรแกรม
- การตรวจดูค่าของตัวแปรต่าง ๆ ในโปรแกรม
- การตรวจดูค่าอย่างละเอียดของตัวแปรที่มีโครงสร้างซับซ้อน เช่น อะเรย์ และ structure
- การเปลี่ยนค่าของตัวแปร

ก่อนจะดีบักให้ตรวจสอบโปรแกรมที่จะดีบักให้แน่ใจว่ามีอยู่ 2 ไฟล์ คือ TCDEMO.C และ TCDEMO.EXE สำหรับตัวอย่างการดีบักภาษาซี หรือ TPDEMO.PAS และ TPDEMO.EXE สำหรับภาษาปาสคาล เริ่มต้นดีบักโดยพิมพ์

TD TCDEMO <ENTER> (สำหรับภาษาซี) หรือ

TD TPDEMO <ENTER> (สำหรับภาษาปาสคาล)



รูปที่ 2 แสดงหน้าจอหลัก (main screen) ของเทอร์โบดีบักเกอร์

เทอร์โบดีบักเกอร์จะโหลดโปรแกรม และให้เคอร์เซอร์อยู่ที่จุดเริ่มต้นของโปรแกรม ดังรูปที่ 2 ซึ่งประกอบด้วยบรรทัดที่เป็นเมนูหลัก หน้าต่างของโมดูลซึ่งเป็นหน้าต่างที่แอคตีฟอยู่(สังเกตจากมีกรอบเป็นเส้นคู่) หน้าต่างของ Watch และบรรทัดล่างสุดซึ่งเป็นคำสั่งที่ใช้ฟังก์ชันคีย์ ถ้าเราหลงทางระหว่างการดีบักให้กด Ctrl-F2 เพื่อโหลดโปรแกรมใหม่โดยที่ลูกศร (>) จะไปอยู่ที่ตำแหน่งเริ่มต้นของโปรแกรมตามเดิม โดยไม่มีการเคลียร์เบรคพอย (breakpoint) หรือวอชต์ (watch) แต่ถ้าต้องการให้เคลียร์เบรคพอยต์และวอชต์ด้วย ให้กด Alt-F L เพื่อโหลดโปรแกรมใหม่ และถ้ากด F3 โดยไม่ตั้งใจหน้าต่างจะถูกปิดไป ให้กด Alt-W U เมื่อใดที่เราต้องการข้อมูลเพิ่มเติมเพื่ออธิบายหน้าต่าง คำสั่ง หรือข้อมูลแสดงความผิดพลาด ให้กด F1 ได้ทันที

จากรูปที่ 2 จะเห็นลูกศร (>) อยู่ในคอลัมน์ซ้ายสุดของหน้าต่างโมดูล แสดงตำแหน่งที่เทอร์โบดีบักเกอร์หยุดโปรแกรมของเราเพราะว่าเรายังไม่ได้รันโปรแกรม ลูกศรจึงหยุดที่ต้นโปรแกรม (main) กด F7 เพื่อติดตามการทำงานทีละบรรทัดของโปรแกรมลูกศรจะไปหยุดอยู่ที่บรรทัดใหม่ซึ่งเป็นบรรทัดที่จะทำการเอ็กซีคิวต์ต่อไป ทางขวาสุดของชื่อโมดูลจะแสดงหมายเลขบรรทัดที่เคอร์เซอร์อยู่

การควบคุมการรันโปรแกรม

เทอร์โบดีบักเกอร์มีวิธีการควบคุมการเอ็กซีคิวต์โปรแกรมอยู่หลายวิธี ซึ่งแสดงอยู่ในเมนู Run ตัวอย่างเช่น สมมติว่าเราต้องการเอ็กซีคิวต์ถึงบรรทัดที่ 38 ให้เลื่อนเคอร์เซอร์ไปที่บรรทัดที่ 38 แล้วกด F4 (F4-Here หมายถึงเอ็กซีคิวต์ถึงตำแหน่งนี้) โปรแกรมจะรันถึงบรรทัดที่ 38 (แต่ยังไม่ทำบรรทัดที่ 38) ต่อไปกด F7 เป็นการเรียกฟังก์ชัน showargs ดังนั้นเคอร์เซอร์(และลูกศร) จะกระโดดไปบรรทัดที่ 150 ทันที ซึ่งเป็นจุดเริ่มต้นของฟังก์ชัน showargs กด F7 ต่อ จะเอ็กซีคิวต์ต่อไปในฟังก์ชัน showargs

โดยทำทีละบรรทัดจนกระทั่ง return เคอร์เซอร์จะกระโดดกลับไปบรรทัดที่ 39 ก่อนที่จะเข้าไปในฟังก์ชัน showargs ขณะที่อยู่บรรทัดที่ 38 ถ้าเรากด F8 แทน F7 เคอร์เซอร์จะมาอยู่บรรทัดที่ 39 เลย ซึ่งก็คือ F8 จะเอ็กซีคิวต์ทีเดียวตลอดฟังก์ชันเลย ซึ่งใช้ในกรณีที่เราแน่ใจแล้วว่าฟังก์ชันนั้นทำงานได้ถูกต้องแล้ว แต่ถ้าเราเผลอกด F7 แทน F8 มันจะเอ็กซีคิวต์เข้าไปในฟังก์ชันนั้น เราสามารถกด Alt-F8 (ซึ่งเป็นการใช้แบบย่อของเมนู Run/Until Return) ซึ่งจะเอ็กซีคิวต์ฟังก์ชันนั้นทั้งหมด จนกระทั่งรีเทอร์นค่ากลับมา โดยไม่ต้องเอ็กซีคิวต์ทีละบรรทัด

ในกรณีที่เราต้องการติดตามขั้นตอน (Trace) ของโปรแกรมไปเรื่อย ๆ แทนที่จะต้องกด F7 ทีละครั้ง เราสามารถใช้เทอร์โบดีบักเกอร์ทำแทนโดยใช้ Alt-F4 (ซึ่งเป็นการใช้แบบย่อของเมนู Run/Animate) มันจะทำคำสั่ง Trace ติดต่อกันไป แต่ละคำสั่งจะห่างกันเป็นช่วงเวลาที่เราสามารถกำหนดเองได้ (เทอร์โบดีบักเกอร์กำหนดให้เท่ากับ 3/10 วินาที) แล้วหยุดการทำงานด้วยการกดคีย์ใดคีย์หนึ่ง

บ่อยครั้งที่เราดีบักโปรแกรมโดยการติดตามขั้นตอนของโปรแกรม หรือการใช้คำสั่งอื่นประกอบก็ตาม เราอาจจะเอ็กซีคิวต์เลยจุดที่เราคิดว่าโปรแกรมผิดพลาด ซึ่งทำให้เราต้องเริ่มโหลดโปรแกรมใหม่ และทำขั้นตอนเดิมที่เราทำมาแล้วซึ่งเป็นสิ่งที่น่าเบื่อมาก เทอร์โบดีบักเกอร์มีแมคโครให้เราใช้โดยสามารถเก็บชุดของการกดคีย์ที่เป็นคำสั่งต่อเนื่องกันไป รวมอยู่ในการกดคีย์เพียงครั้งเดียว ตัวอย่างเช่นเมื่อเราโหลดโปรแกรมแล้ว เริ่มต้นการเก็บชุดของการกดคีย์โดยกด Alt-= แล้วกดคีย์ที่ต้องการจะกำหนดให้เป็นคีย์แมคโคร (อย่าใช้คีย์ที่ซ้ำกับคำสั่งของเทอร์โบดีบักเกอร์) เช่น shift-F1 ถ้าคีย์นี้เคยถูกกำหนดมาก่อนแล้ว จะมีข้อความถามว่าให้เก็บทับของเดิมหรือไม่ มุมขวาของหน้าต่างที่ 1 จะปรากฏ RECORDING ให้เริ่มดีบักโปรแกรมไปจนกระทั่งถึงจุดที่คิดว่าโปรแกรมจะผิดพลาดแล้วกด Alt--(Alt กับเครื่องหมายลบพร้อมกัน) ข้อความ RECORDING จะหายไป เป็นอันสิ้นสุดการเก็บคีย์ของแมคโคร แล้วดีบักโปรแกรมต่อไป เมื่อจะเริ่มต้นดีบักใหม่ให้ถึงจุดเดิมนี้ ก็เพียงแต่กดคีย์ shift-F1 เท่านั้น

อีกวิธีหนึ่งจะให้โปรแกรมเอ็กซีคิวต์ถึงตำแหน่งที่ต้องการ โดยกำหนดฟังก์ชัน หรือหมายเลขบรรทัด โดยไม่ต้องเลื่อนเคอร์เซอร์ไปที่ตำแหน่งนั้น เช่น กด Alt-F9 แล้วพิมพ์ readaline <ENTER> โปรแกรมจะรันแล้วหยุดที่จุดเริ่มต้นของฟังก์ชัน readaline หรือใส่หมายเลขบรรทัดโดยพิมพ์ #100 <ENTER> โปรแกรมจะรันแล้วหยุดที่บรรทัดที่ 100

การตั้งเบรคพอยต์เป็นอีกวิธีหนึ่งในการควบคุมการหยุดโปรแกรม โดยการเลื่อนเคอร์เซอร์ไปยังบรรทัดที่ต้องการให้หยุดรันโปรแกรม แล้วกด F2 บรรทัดนั้นปรากฏไฮไลต์ กด F9 เพื่อรันโปรแกรม โปรแกรมจะรันถึงบรรทัดที่ตั้งเบรคพอยต์ไว้ เบรคพอยต์สามารถเอาออกได้โดยการกด F2 อีกครั้งหนึ่งเมื่อเคอร์เซอร์อยู่ที่ตำแหน่งที่ตั้งเบรคพอยต์ไว้ รายละเอียดของการใช้เบรคพอยต์สามารถดูจากเมนู Breakpoint และใช้ help ประกอบ (กด F1)

การดูค่าตัวแปร (watching)

เมื่อเราทราบวิธีควบคุมการรันโปรแกรมแล้ว ต่อไปเราจะมาดูว่าการดูค่าของตัวแปร เช่น ตัวแปรเลขจำนวนเต็ม (integer) สตริง (string) ทำอย่างไร หน้าต่าง Watches (หมายเลข 2) ซึ่งอยู่ด้านล่างของหน้าจอใช้แสดงค่าของตัวแปรที่ต้องการจะดู ตัวอย่างเช่น ต้องการดูค่าของตัวแปรชื่อ nwords ให้เลื่อนเคอร์เซอร์ไปที่ชื่อตัวแปรที่บรรทัดที่ 41 แล้วกด Ctrl-W (ซึ่งเป็นการใช้แบบย่อของ local menu command : Alt-F10 W) ตัวแปร nwords จะปรากฏที่หน้าต่างของ watches รวมทั้งชนิด (type) และค่าของมัน ขณะที่เอ็กซีคิวต์โปรแกรมถ้าค่าของตัวแปรนี้เปลี่ยนไป ค่าที่ปรากฏบนหน้าต่าง watches จะเปลี่ยนตามไปด้วย

ในหน้าต่างของ watches มีคำสั่งอะไรบ้าง ลองกด Alt-2 เพื่อแอคตีฟหน้าต่างที่ 2 แล้วกด Alt-F10 ดู จะเห็นว่ามีคำสั่งภายในหน้าต่าง watches ดังรูปที่ 3 ซึ่งมีรายละเอียดดังนี้

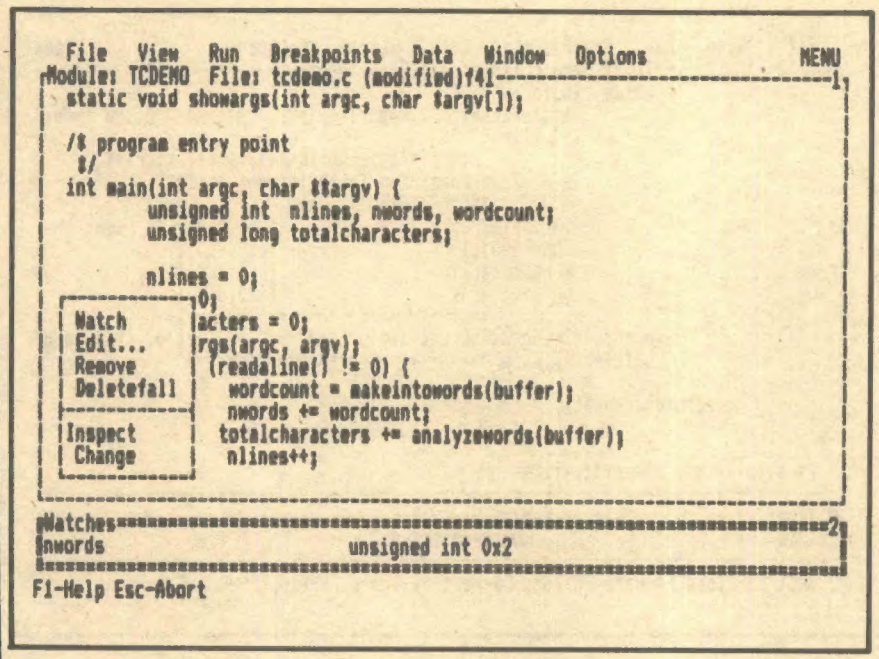
Watch ดูค่าตัวแปรหรือนิพจน์ (โดยการพิมพ์ชื่อตัวแปรหรือนิพจน์ลงไป)

Edit แก้ไขตัวแปรหรือนิพจน์เดิมในหน้าต่าง Watches

Remove เอานิพจน์หรือตัวแปรตัวล่าสุดในหน้าต่าง Watches ออก

Delete ALL เอานิพจน์หรือตัวแปรในหน้าต่าง Watches ออกทั้งหมด

Inspect เปิดหน้าต่าง Inspect เพื่อแสดงรายละเอียดของตัวแปรที่ไฮไลต์ในหน้าต่าง Watches ซึ่งตัวแปรนั้นอาจจะเป็นตัวแปรที่มีโครงสร้างซับซ้อน เช่น อะเรย์ ยูเนียน หรือ structure เพื่อดูสมาชิก



รูปที่ 3 แสดงคำสั่งภายในของหน้าต่าง Watches

ทุก ๆ ตัวของมัน

Change เปลี่ยนค่าตัวแปรที่ไฮไลต์ในหน้าต่าง Watches

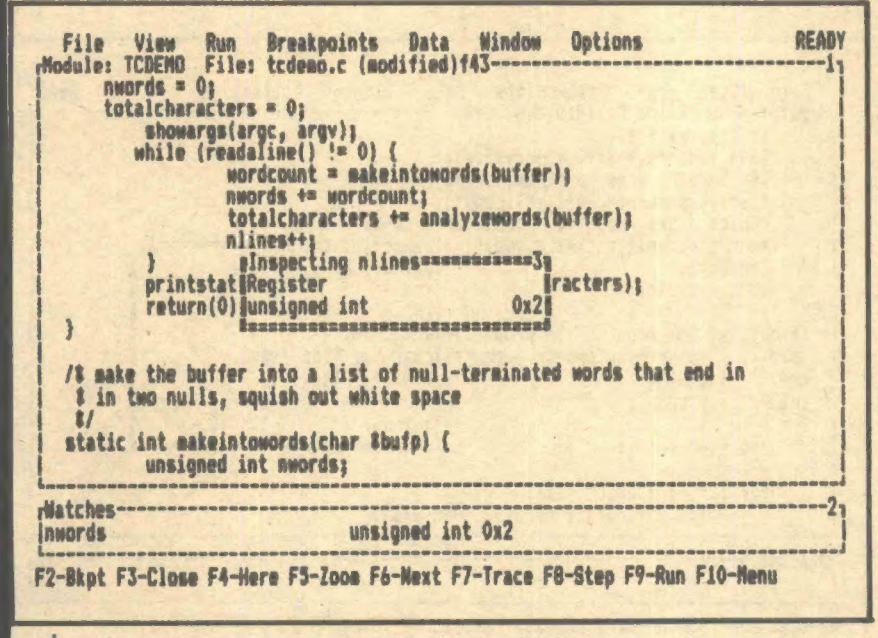
การตรวจดูข้อมูลอย่างละเอียด (Inspecting)

ขณะที่เราหยุดโปรแกรม มีหลายวิธีที่จะดูข้อมูลโดยการใช้คำสั่ง Inspec (ซึ่งอยู่ใน local menu command และใน Data menu) สมมติว่าเราต้องการดูตัวแปรชื่อ nlines ก็เลื่อนเคอร์เซอร์ไปที่ตัวแปร nlines แล้วกด Ctrl-I หน้าต่างของการ Inspect จะปรากฏขึ้นมาดังรูปที่ 4

ในหน้าต่างของการ Inspect (รูปที่ 4) บรรทัดแรก แสดงว่าตัวแปรนี้ถูกกำหนดให้เก็บค่าในรีจิสเตอร์ของซีพียู บรรทัดที่ 2 แสดงชนิดของตัวแปรและค่าของตัวแปร ปิดหน้าต่างโดยกด F3 คราวนี้ลองมาดูตัวแปรที่มีโครงสร้างข้อมูลแบบอะเรย์บ้าง เลื่อนเคอร์เซอร์ไปที่ตำแหน่งตัวแปรชื่อ letterinfo [] แล้วกด Ctrl-I จะปรากฏหน้าต่างขึ้น บรรทัดแรกต่อจากชื่อหน้าต่างจะแสดงที่อยู่ในหน่วยความจำหลักของสมาชิกตัวแรกของตัวแปรอะเรย์ บรรทัดต่อมาแสดงหมายเลขสมาชิกและค่าของมัน โดยสามารถใช้เป็นลูกศรและแป้น Pg Up และ Pg Dn เพื่อเลื่อนดูสมาชิกตัวต่อ ๆ ไป ส่วนบรรทัดล่างสุดจะแสดงชนิดของอะเรย์ ในที่นี้คืออะเรย์ของ structure ที่ชื่อ linfo ดังแสดงในรูปที่ 5 เลื่อนแถบของเมนูไปยังสมาชิกตัวใดตัวหนึ่ง สมมติว่าหมายเลข 3 กด Enter จะปรากฏหน้าต่างที่ 4 แสดงส่วนประกอบของข้อมูลแบบ structure ว่าประกอบด้วยตัวแปรอะไรบ้าง ในที่นี้คือตัวแปร count และ firstletter ซึ่งเป็นชนิดตัวเลขจำนวนเต็มไม่คิดเครื่องหมาย (unsigned int) ทั้งคู่ การตรวจดูข้อมูลนั้น รายละเอียดของข้อมูลขึ้นกับชนิด (type) ของตัวแปร ซึ่งแบ่งได้เป็น

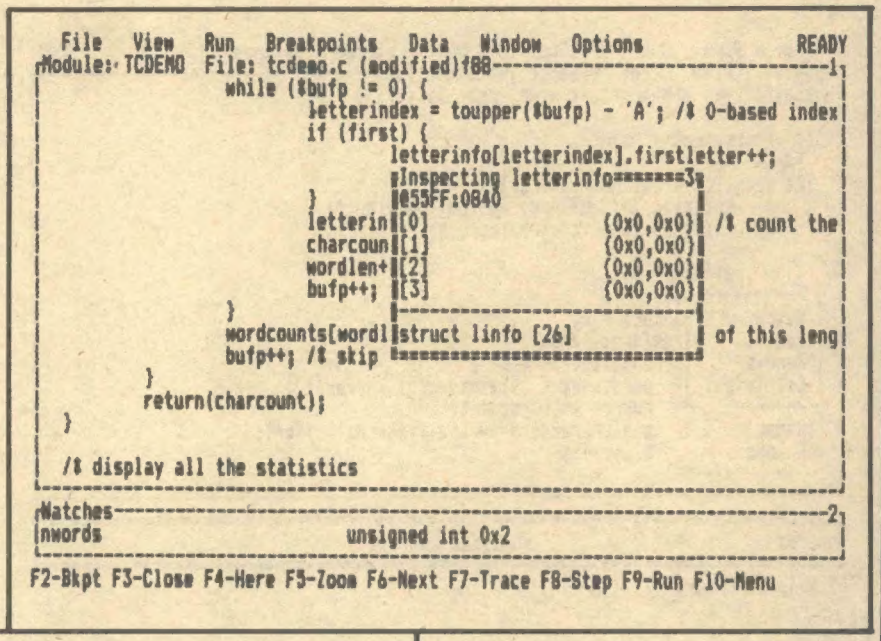
1. สเกลาร์ (scalar) ซึ่งจะทำให้รายละเอียดเพียงที่อยู่ในหน่วยความจำหลัก ชนิดของตัวแปรและค่าของตัวแปรเท่านั้น

2. พอยน์เตอร์ (pointer) จะแสดงที่อยู่ของตัวแปรชนิดพอยน์เตอร์ ซึ่งแสดงดังรูปที่ 6 ในบรรทัดแรก 547D: 0FD6 เป็นที่อยู่ของตัวแปร keys ซึ่งอยู่ในรูป stack segment : stack pointer 5419 เป็นค่าของ data segment และ 0156 เป็นค่าออฟเซตซึ่งเก็บอยู่ใน SS : SP ส่วนบรรทัดต่อมา 'a'...'s' เป็นตัวอักษรที่เก็บอยู่ใน data segment ที่มีค่าออฟเซตตั้งแต่ 0156-015B ตาม

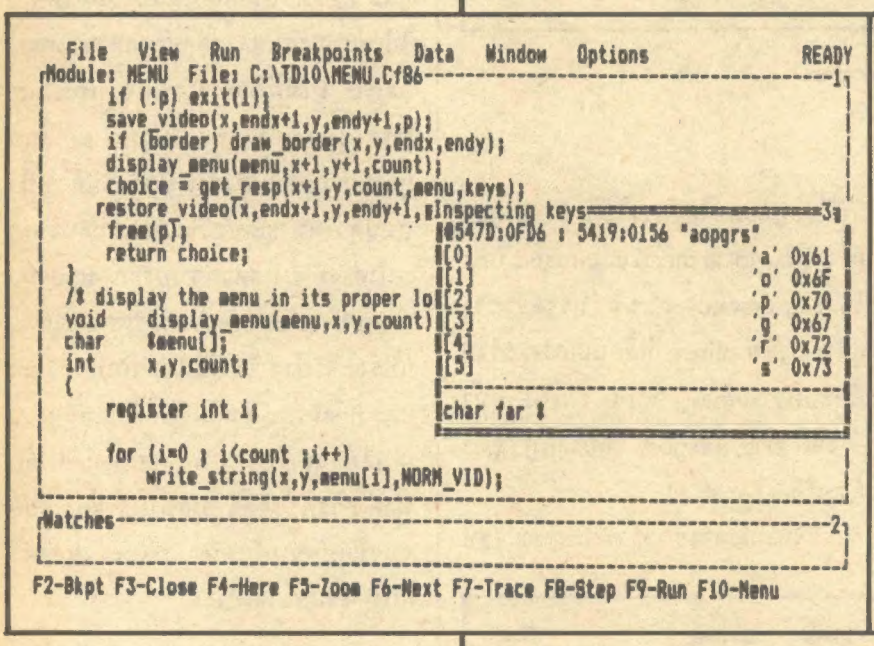


รูปที่ 4 แสดงหน้าต่างของการ Inspect ตัวแปร nlines

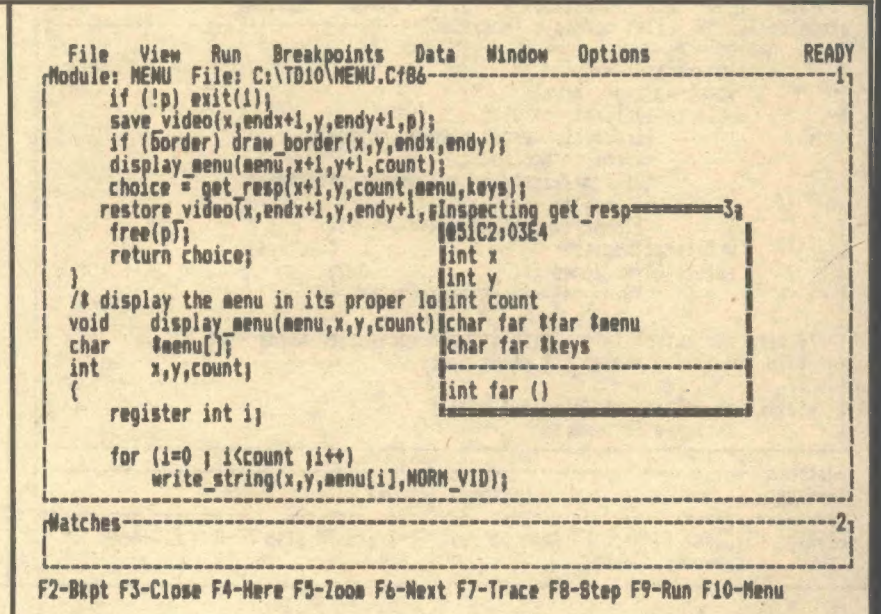
รูปที่ 5 แสดงหน้าต่างของการ Inspect ตัวแปรอะเรย์ letterinfo



รูปที่ 6 แสดงหน้าต่างของการ Inspect ตัวแปรชนิดพอยน์เตอร์ 4 ช่อ Keys



รูปที่ 7 แสดงหน้าต่างของการ Inspect ตัวแปรที่เป็นฟังก์ชัน



ลำดับ ค่าต่าง ๆ เหล่านี้สามารถดูรายละเอียดได้จากเมนู View/CPU

3. อะเรย์ (array) ซึ่งได้กล่าวไปแล้วเมื่อสักครู่นี้

4. ยูเนียน (union) และ structure จะแสดงสมาชิกแต่ละตัวและค่าของตัวแปร ส่วนบรรทัดล่างสุดจะแสดงชนิดของข้อมูลของสมาชิก

5. ฟังก์ชัน (function) จะแสดงชื่อของฟังก์ชัน พารามิเตอร์เมื่อเรียกใช้ฟังก์ชัน และชนิดของข้อมูลที่ส่งกลับจากฟังก์ชัน ดังแสดงในรูปที่ 7 ปิดหน้าต่างของการ Inspect ทั้งหมดโดยกด F3 แต่ถ้าจะถอยกลับทีละหน้าต่างให้กด ESC ถ้าต้องการกลับไปหน้าต่างโมดูลและจะกลับมาที่หน้าต่างของการ Inspect อีก ควรใช้การกด Alt พร้อมกับหมายเลขหน้าต่าง เช่น Alt-1 กลับไปหน้าต่างโมดูล และ Alt-3 หรือ Alt-4 กลับมาหน้าต่าง Inspect อีก เทอร์โบดีบักเกอร์ให้คุณเปิดหน้าต่างได้พร้อมกัน 9 หน้าต่าง แต่จะแอคตีฟเพียงทีละหน้าต่างเดียวเท่านั้น แต่ละหน้าต่างสามารถเคลื่อนย้ายและเปลี่ยนขนาดได้ โดยการกด Scroll และใช้แป้นลูกศร หรือแป้นลูกศรควบคู่กับ shift เพื่อย้ายหรือเปลี่ยนขนาดของหน้าต่างที่กำลังแอคตีฟอยู่ตามลำดับ

การเปลี่ยนค่าของตัวแปร

เมื่อเราสามารถดูข้อมูลของตัวแปรต่าง ๆ แล้ว คราวนี้ลองมาดูการเปลี่ยนค่าของตัวแปรเพื่อดูผลกระทบต่อโปรแกรม ขณะที่อยู่ในหน้าต่าง Inspect กด Alt-F10 เพื่อแสดง local menu กด C จากนั้นพิมพ์นิพจน์ใด ๆ ก็ตามในภาษาซี (สำหรับผู้ที่ใช้ปาสคาลก็พิมพ์นิพจน์ในภาษาปาสคาล) ซึ่งแสดงจำนวนได้แล้ว Enter ค่าที่แสดงในหน้าต่างจะเปลี่ยนตามไปด้วย หรืออาจจะใช้ Alt-D E แล้วพิมพ์ตัวแปรที่ต้องการจะเปลี่ยนลงไป

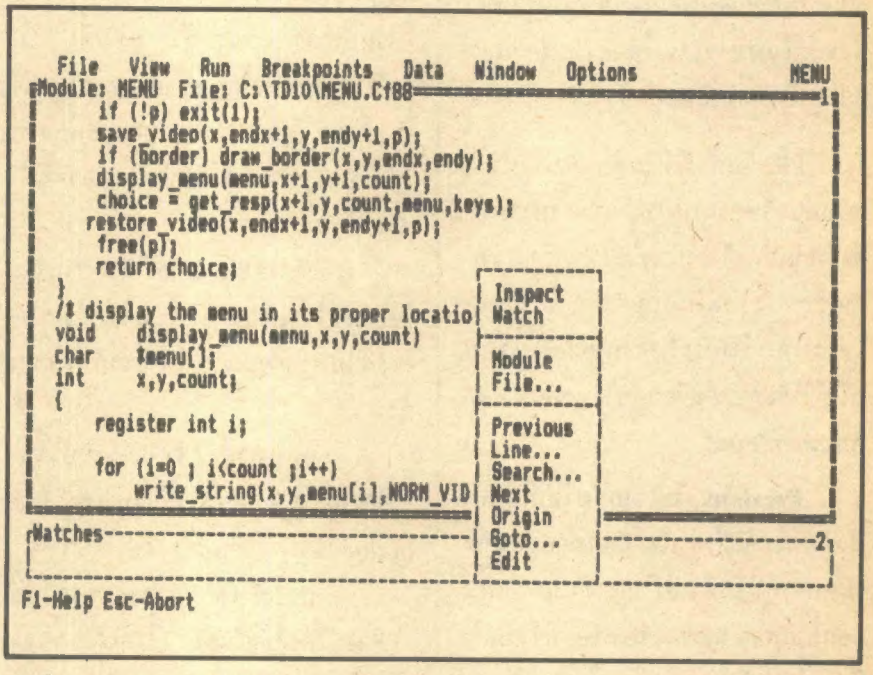
การใช้คำสั่งจากเมนูภายใน (local menu command)

เมื่ออยู่ในหน้าต่างโมดูล เทอร์โบดีบักเกอร์มีคำสั่งให้ใช้เพื่อช่วยในการดีบักโปรแกรมโดยกดปุ่ม Alt-F10 จะปรากฏเมนูขึ้นดังรูปที่ 8

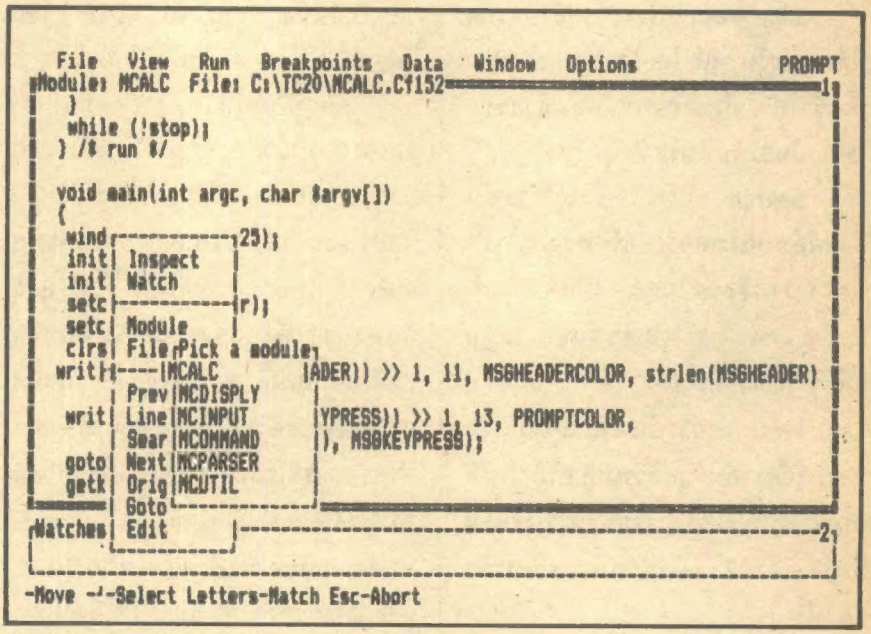
จะเห็นว่ามีคำสั่งต่าง ๆ ซึ่งมีรายละเอียดดังต่อไปนี้

Inspect และ **watch** ซึ่งได้กล่าวไปแล้วตอนต้น

Module ในกรณีที่โปรแกรมของเราสร้างมาจากโปรแกรมย่อย (module)



รูปที่ 8 แสดงเมนูภายใน (local menu command)



รูปที่ 9 แสดงเมนูในการใช้คำสั่ง module

หลายโมดูล เราสามารถจะเรียกซอร์สโปรแกรมจากโมดูลอื่น (ที่ไม่ได้ปรากฏบนจอในขณะนั้น) มาดูได้โดยคำสั่งโมดูล ดังแสดงในรูปที่ 9

โมดูลที่จะแสดงอยู่บนจอคือ MC-PARSER เราสามารถดึงเอาโมดูลอื่น เช่น MCALC ออกมาดูได้ทีละโมดูล แต่ถ้าต้องการดูมากกว่าหนึ่งโมดูลในขณะเดียวกัน ให้ใช้คำสั่ง View/Another/Module เพื่อเปิดอีกหน้าต่างหนึ่ง ซึ่งจะเป็นหมายเลขที่ต่อจากหมายเลขของหน้าต่างที่เปิดอยู่แล้ว

File ในกรณีที่โมดูลของเราประกอบด้วยไฟล์หลายไฟล์ เราสามารถจะดึงไฟล์แต่ละไฟล์มาดูได้ โดยเลือกจากรายชื่อของไฟล์ในเมนู ถ้าคุณต้องการดูไฟล์มากกว่าหนึ่งไฟล์ในขณะเดียวกัน ให้ใช้คำสั่งทำนองเดียวกับโมดูลคือ View/Another/File

Previous ให้เรากลับไปเพื่อดูคำสั่งที่ผ่านมา ถ้าคำสั่งที่ผ่านมาอยู่คนละโมดูล มันจะดึงเอาโมดูลก่อนหน้านั้นออกมาแสดง และเคอร์เซอร์อยู่ที่ตำแหน่งที่จะทำคำสั่ง แล้วทำให้มาอยู่โมดูลปัจจุบัน

Line เลื่อนเคอร์เซอร์ไปยังบรรทัดที่ต้องการในไฟล์ โดยใส่หมายเลขบรรทัดเข้าไป ถ้าใส่เกินบรรทัดสุดท้ายของไฟล์ มันจะไปที่บรรทัดสุดท้าย

Search ค้นหาสตริงโดยเริ่มต้นตั้งแต่ตำแหน่งเคอร์เซอร์ขณะนั้น ถ้าต้องการหาตลอดโมดูล ให้กด Ctrl-PgUp เพื่อไปยังจุดเริ่มต้นของโมดูลแล้วจึงทำการค้นหา

Next ทำการค้นหาตัวถัดไป ในบางกรณีเราค้นหาแล้วพบตัวที่มีส่วนที่เหมือนกับที่เราต้องการหา แต่ไม่ใช่ตัวที่ต้องการคำสั่ง nextจะทำการค้นหาตัวถัดไป

สำหรับผู้ที่ใช้โปรแกรมภาษาปาสคาลก็ในทำนองเดียวกัน จะต่างกันก็เฉพาะรูปแบบและชนิดของข้อมูลเท่านั้น ต่อไปเราจะมาดูว่าเทอร์โบดีบักเกอร์มียูทิลิตี้อะไรบ้าง

ยูทิลิตี้ของเทอร์โบดีบักเกอร์

มียูทิลิตี้หลายตัวซึ่งมีบทบาทมากต่อเทอร์โบดีบักเกอร์ หลังจากที่เราดีบักโปรแกรมเรียบร้อยแล้ว ยูทิลิตี้ TDSTRIP จะเอาข้อมูลซึ่งเป็นตารางของสัญลักษณ์ (symbols table) ออกจากไฟล์ EXE ซึ่งทำให้ไม่จำเป็นต้องคอมพายล์โปรแกรมใหม่

ยูทิลิตี้ TDMAP ทำให้เราสามารถใช้เทอร์โบดีบักเกอร์ดีบักโปรแกรมเทอร์โบปาสคาล 4.0 ได้ โดยมีขั้นตอนดังนี้

– คอมพายล์โปรแกรมเทอร์โบปาสคาล 4.0 ด้วย command line option/$T+ เพื่อสร้างไฟล์ .TPM

– สร้างไฟล์ .MAP จากไฟล์ .TPM โดยใช้ยูทิลิตี้ TDMAP ของเทอร์โบปาสคาล 4.0

– รัน TDMAP เพื่อรวมไฟล์ .EXE กับข้อมูลจากไฟล์ .MAP ให้อยู่ในรูปแบบของเทอร์โบดีบักเกอร์

อีกยูทิลิตี้หนึ่งที่ควรรู้จักคือ TDUMP เป็นตัววิเคราะห์ไฟล์ซึ่งจะทำการกระจายโครงสร้างของไฟล์ .EXE, .OBJ และ .LIB TDUMP จะถอดรหัสเฮดเดอร์ (header) ของไฟล์ .EXE และแสดงการให้ค่าแอดเดรสเริ่มต้นของสแต็กเซกเมนต์ ทางเข้าของโปรแกรม และแอดเดรสต่าง ๆ ที่โหลดเดอร์ต้องการสำหรับการรีโลเคต สำหรับออปเจกไฟล์ TDUMP จะฟอร์แมตข้อมูลในการกำหนดเซกเมนต์ สัญลักษณ์ของ PUBLIC และแอดเดรสต่าง ๆ ที่จะส่งให้ลิงค์เกอร์ TDUMP ยังสามารถแสดงค่าของข้อมูลและโค้กเซกเมนต์ในรูปแบบของเลขฐานสิบหก สำหรับไฟล์ .LIB TDUMP ก็จะแสดงเช่นเดียวกับไฟล์ .OBJ และข้อมูลเฉพาะบางอย่างของไลบรารี

ถ้าข้อมูลไม่ได้อยู่ใน 3 รูปแบบนี้ TDUMP จะแสดงข้อมูลในไฟล์เป็นเลขฐานสิบหก หรือรหัสแอสกีหรือทั้ง 2 อย่าง TDUMP ไม่สามารถวิเคราะห์โครงสร้างของไฟล์ในเทอร์โบปาสคาล 5.0 ซึ่งมีนามสกุลเป็น .TPU หรือ .TPL ซึ่งเทียบเท่ากับไลบรารีในภาษาอื่น และไม่มียูทิลิตี้ใดในเทอร์โบดีบักเกอร์ที่สามารถทำงานเช่นนี้ได้

TDPACK เป็นยูทิลิตี้สุดท้ายที่เราจะมาทำความรู้จัก TDPACK เป็นยูทิลิตี้ที่ใช้ลดขนาดของข้อมูลที่ใช้ในการดีบักซึ่งเพิ่มเข้าไปในไฟล์ .EXE ตัวอย่างเช่น ไฟล์ขนาด 55 กิโลไบต์ เมื่อคอมพายล์โดยรวมข้อมูลในการดีบักจะเพิ่มขนาดเป็น 115 กิโลไบต์ TDPACK สามารถกำจัดข้อมูลที่ซ้ำกัน เช่น สตริง และชนิดของข้อมูล ได้ประมาณ 9,000 ไบต์ ถ้าเทอร์โบดีบักเกอร์ใช้หน่วยความจำจนหมดและยังไม่พอ ในการดีบักโปรแกรมขนาดใหญ่การรัน TDPACK สามารถช่วยได้

ส่งท้าย

จากตัวอย่างที่ผมยกขึ้นมาให้ดูนั้น เป็นเพียงส่วนหนึ่งของการใช้เทอร์โบดีบักเกอร์เท่านั้น ยังมีรายละเอียดอื่น ๆ อีกมากที่มิได้กล่าวถึง เช่น การดีบักแบบ remote การดีบักบนเครื่อง 80386 การตั้งเบรคพอยต์แบบมีเงื่อนไขการดีบักในระดับแอสเซมบลี ซึ่งผู้อ่านสามารถหาได้จากหนังสือ TURBO DEBUGGER User's Version 1.0 หวังว่าทุกท่านคงจะสนุกกับการดีบักโปรแกรมด้วยเทอร์โบดีบักเกอร์นะครับ